# SOFTWARE ARCHITECHTURE FOR ACADEMIC AUDIT INFORMATION MANAGEMENT SYSTEM

NOR AZIMA BINTI ABU SEMAN

UNIVERSITI UTARA MALAYSIA

2010

# SOFTWARE ARCHITECHTURE FOR ACADEMIC AUDIT INFORMATION MANAGEMENT SYSTEM

**A thesis submitted to the College of Art and Sciences in partial fulfillment of the requirements for the degree Master of Science (Information Technology)**
**Universiti Utara Malaysia**

**By**

**NOR AZIMA BINTI ABU SEMAN**
**MSc(IT)**

**College of Arts and Sciences**
**06010 UUM Sintok, Kedah**
**s89451@uum.edu.my**

**KOLEJ SASTERA DAN SAINS**
**(College of Arts and Sciences)**
**Universiti Utara Malaysia**

***PERAKUAN KERJA KERTAS PROJEK***
***(Certificate of Project Paper)***

Saya, yang bertandatangan, memperakukan bahawa
*(I, the undersigned, certify that)*

**NOR AZIMA ABU SEMAN**
**(89451)**

calon untuk Ijazah
*(candidate for the degree of)* **MSc. (Information Technology)**

telah mengemukakan kertas projek yang bertajuk
*(has presented his/her project paper of the following title)*

**SOFTWARE ARCHITECTURE FOR ACADEMIC AUDIT INFORMATION MANAGEMENT SYSTEM**

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
*(as it appears on the title page and front cover of project paper)*

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan.
*(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).*

Nama Penyelia Utama
*(Name of Main Supervisor)*: **ASSOC. PROF. NAZIB NORDIN**

Tandatangan
*(Signature)* : _____

Tarikh
*(Date)* : 18/5/10

## PERMISSION TO USE

In presenting this thesis, in partial fulfillment of the requirements for a post graduate degree from Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor or, in their absence, by the Dean of Academic Office. It is understood that any copying or publications, or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Request for permission to copy or make other use of materials in this thesis in whole or in part should be addressed to:

Dean of Academic Office
College of Arts and Sciences
Universiti Utara Malaysia
06010 Sintok
Kedah Darul Aman

## ABSTRACT

The main purpose of this research is to give an understanding of how important is software architecture in a system development process. Software architecture is the second phase in system development life cycle which gave a huge contribution of the system fundamentals that determines the system's remaining development, its deployment and maintenance life. The relationship among requirements, processes, users and results are a never ending relationship where they complete each other and work as a unit. It is much like a framework of ideas, concepts and requirements that are designed before developing and carrying out the actual actions. A comprehensive software architecture which meets the quality standards can be used and referred to for other system's development which also shares the same fundamental concepts. Organization management and governing processes such as auditing management requires a comprehensive and efficient architecture to ensure the auditing processes are done accurately based on the standards set up by the quality control organization. Malaysian Qualifications Agency (MQA) is one of the organizations which monitors and coordinate both the quality assurance and accreditation of national higher learning institute. MQA auditing processes are carried out based on the international quality assurance processes and standards. To ensure the accuracy of auditing processes, the Architecture Tradeoff Analysis Method (ATAM) has been chose as an assessment tool to test the correctness of the architecture.

# ABSTRAK

Tujuan utama projek ini adalah memahami betapa pentingnya senibina sesuatu sistem di dalam proses pembangunan sistem. Senibina sistem terkandung di dalam fasa kedua di dalam kitar hayat pembangunan sistem yang mana memberi impak besar terhadap asas pembangunan dan penyelenggaraan sesuatu sistem. Hubungan di antara keperluan sistem, proses, pengguna sistem dan sistem itu sendiri tiada penghujungnya kerana setiap subjek tersebut saling memerlukan dan menyempurnakan diantara satu sama lain sebagai satu pasukan. Merujuk kepada senibina perisian, ia merupakan kerangka asas daripada idea, konsep dan keperluan bagi sesuatu sistem sebelum ianya dibangunkan dan dilaksanakan menjadi realiti. Senibina sistem yang komprehensif adalah yang memenuhi standard kualiti yang ditetapkan yang boleh menjadi rujukan kepada pembangunan sistem lain yang berasaskan konsep yang sama. Pengurusan di dalam organisasi yang melibatkan pengurusan kualiti seperti pengurusan audit memerlukan satu sistem yang komprehensif dan efisyen bagi memastikan proses mengaudit dapat dilaksanakan dengan tepat dan berkesan berlandaskan standard yang ditetapkan oleh agensi kawalan kualiti. Agensi Kelayakan Malaysia (MQA) adalah sebuah organisasi yang dipertanggungjawabkan untuk mengurus dan memantau jaminan kualiti dan akreditasi bagi Institut Pengajian Tinggi Awam. MQA melaksanakan proses mengaudit berasaskan kepada standard dan proses jaminan kualiti antarabangsa. Bagi memastikan sistem yang akan dibangunkan adalah komprehensif, satu poendekatan akan dilaksanakan terhadap senibina sistem ini dengan menggunakan *Architecture Tradeoff Analysis Method (ATAM)* sebagai satu kaedah penilaian terhadap sebinina sistem.

## ACKNOWLEDGEMENT

Alhamdulillah, all praise to Allah. I am very grateful for having this opportunity to thanks several individuals that have been helping me not only through out finishing this thesis, but for all those years that I have been struggling to conclude this study.

First of all my thousand appreciations to all my lecturers especially Assoc. Prof. Madya Nazib Bin Nordin for being my supervisor and has been guiding with patience and criticism until this thesis is finally prepared and may Allah cherish your lives with cheer and happiness.

Second, I would like to thank my employer, Mr. Nasrudin Bin Abd. Shukor for fulfilling my desire and approved me to continue in this study. Your understanding is much appreciated. Also no forgotten to Mr. Fauzi Bin Baharom, officer of Quality Management Unit and Ir. Salmah, Academic Deputy Dean of School of Material because of their cooperated in explain about audit process with extremely sincere.

My utmost gratitude to all my family members which has been there for me and giving me spirit to pursue on this study; especially for my daddy and mummy, Haji Abu Seman Bin Haji Ahmad and Hajah Faridah Binti Haji Husin. And not to forget my beloved husband, Rhafizuan Bin Rusli, thanks for you're endlessly support. Not to forget too, may beloved son, Muhammad Riffat Raidi, your smile make me strong. I could not imagine how my life goes on without having both of you at my side.

Last but not least thanks to my friends, Madalatun-Najah Binti Mahmud, whom have been my loyal advisor through out all semesters, I can not imagine pursuing this study without yours company.

Thank you all.

# CONTENTS

**ITEM** **PAGE**

# CHAPTER 1

# INTRODUCTION

## 1.1 Background of the Project

This project is initiated to fulfill the STIZ6996 course project as one of graduation requirements for Master of Science in Information Technology (MSc. IT) in University Utara Malaysia. The main purpose of this project is to develop software architecture for Academic Audit Information Management System (AAIMS) based on University Malaysia Perlis (UniMAP) practices. Academic audit process is part of preparation process of getting accreditation for UniMAP's academic program from Malaysia Quality Assurance (MQA). The system is a part of the university's internal application system that is known as University Management Information System.

UniMAP's Quality Management Unit (QMU) is the responsible department that manages, coordinate and organize activities of quality management system according to MQA. Specifically, it is responsible to monitor higher educational services and infrastructures that are provided for fulltime Bachelor's Degree Program (Hons) offered in Universiti Malaysia Perlis. It also serve as a two headed spear where by implementing the Quality Management System MS ISO 9001:2008 it also supports the university agenda to achieve UniMAP's quality policy's and objectives. Today, UniMAP is striving to get the *Swa-akreditas* status. This status qualifies all Higher Education Institution accreditation study programmes without having to get approval from MQA and the Ministry of Higher Education (MoHE). Yet, they still have to inform MQA and MoHE about the accreditation process for continuous monitoring and auditing by MQA.

The contents of the thesis is for internal user only

## REFERENCES


Admodisastro, N. and G. Kotonya (2008). Architectural Analysis Approaches: A Component-Based System Development Perspective. High Confidence Software Reuse in Large Systems: 26-38.

Angelov, S., Trienekens, J., and Grefen, P. (2008). Towards a Method for the Evaluation of Reference Architectures: Experiences from a Case. Software Architecture: 225-240.

AusAID (2002). Framework for Financial Systems Assessment. Australian Agency for International Development. Retrieved from http://www.ausaid.gov.au/ngos/pdfs/framework_fin_systems.pdf

Antony, T. (2009). Software Architecture Design Reasoning: A Case for Improved Methodology Support. H. Jun and V. Rajesh. **26**: 43-49.

Asri, Yassin (2004). Model Base Generation of a 3 Tier application. Mattic Software BV, University of Amsterdam.

Azim Sharifloo, A., Saffarian, A. S., and Shams, F. (2009). Two Architectural Practices for Extreme Programming. Advances in Computer Science and Engineering: 964-967).

Bass, L, Clements, P. and Kazman, R. (1999). Software Architecture in Practice, Addison-Wesley Longman Publishing Co., Inc.

Bedir, T. (2004). ASAAM: Aspectual Software Architecture Analysis Method.

Bengtsson, P., and Bosch, J. (2000). "An experiment on creating scenario profiles for software change." Annals of Software Engineering **9**(1): 59-78.

Bosch, J. (2004). Architecture-Centric Software Engineering. Software Product Lines: 22-24.

Budgen, D. (2006). Chapter 1: Software Quality Management [PowerPoint slides]. Retrieved from UVic Department of Electrical and Computer Engineering website: http://www.ece.uvic.ca/~itraore/seng426-06/notes/qual06-1-1.pdf

Byrnes, C and Kyratzoglou, I (2007). Applying Architecture Tradeoff Assessment Method (ATAM) as Part of Formal Software Architecture Review. The MITRE Corporation.

Clements, P., Kazman, R. and Klien, M. (2002). Evaluating Software Architectures: Methods and Case Studies, Addison-Wesley Longman Publishing Co., Inc.

Clements, P., Bass, L., Kazman, R. and Abowd, G. (1995). Predicting Software Quality by Architecture-Level Evaluation. Proceedings of the Fifth International Conference on Software Quality, Austin, Texas.

David, B., J, et al. (2006). Model driven security: From UML models to access control infrastructures, ACM. **15:** 39-91.

David, G. (2000). Software architecture: a roadmap. Proceedings of the Conference on The Future of Software Engineering. Limerick, Ireland, ACM.

Hall, M. (2010). Integrating Servlets and Jsp: The Model View Controller (MVC) Architecture. Prentice Hall and Sun Microsystems Press. J2EE training from website : http://courses.coreservlets.com/

Jackson, N. (1996). "Internal academic quality audit in UK higher education: part I – current practice and conceptual frameworks." Quality Assurance in Education **4**(4): 37–46.

Jacqueline, A. B. (2004). A Critical Evaluation of Academic Internal Audit. Quality Assurance in Education **12**(3): 128-135

Jakob Eyvind, B. (2004). Architectural Prototyping: An Approach for Grounding Architectural Design and Learning.

Jihyun, L., Sungwon, K., and Chang-Ki, K. (2009). "Software Architecture Evaluation Methods Based On Cost Benefit Analysis And Quantitative Decision Making." Kluwer Academic Publishers **14**: 453-475.

John, M. J. (2007). Academic Audit Manual for Use in Cycle 4 Academic Audits, New Zealand Universities Academic Audit Unit.

Jon, (2001). Information Systems Security Auditing. Joint Information Systems Security Audit Initiative Management Planning Guide Committee.

Jonathan, W. (2006). Reconstructing Software Architecture Documentation for Maintainability. University of Amsterdam, Amsterdam, The Netherlands.

Kazman, R. (1998). The Architecture Tradeoff Analysis Method.

Kazman, R. and S. J. Carrière (1999). "Playing Detective: Reconstructing Software Architecture from Available Evidence." Automated Software Engineering **6**(2): 107-138.

Kazman, R., M. Klein, et al. (1999). "Evaluating software architectures for real-time systems." Annals of Software Engineering **7**(1): 71-93.

Kazman, R., Carrière, S., and Woods, S. (2000). Toward a discipline of scenario□based architectural engineering. Annals of Software Engineering **9**(1): 5-33.

Kazman, R., Bass, L., Klein, M., Lattanze, T., and Northrop, L. (2005). "A Basis for Analyzing Software Architecture Analysis Methods." Software Quality Journal **13**(4): 329-355.

Keller, R., Bédard, J.-F., and Saint-Denis, G. (2001). Design and Implementation of a UML-Based Design Repository. Advanced Information Systems Engineering: 448-464.

Kizza, J. M. (2009). Standardization and Security Criteria: Security Evaluation of Computer Products. A Guide to Computer Network Security: 351-363.

Land, R. (2002). A Brief Survey of Software Architecture. Malardalen Real-Time Research Center (MRTC) Report, Department of Computer Engineering, Malardalen University, Västerås, Sweden: 1-15.

Liliana, D. (2002). "A Survey on Software Architecture Analysis Methods." IEEE Transactions on Software Engineering **28**: 638-653.

Lindvall, M., R. T. Tvedt, et al. (2003). "An Empirically-Based Process for Software Architecture Evaluation." Empirical Software Engineering **8**(1): 83-108.

Mario, R. B. (2003). Software Quality Attributes and Architecture Tradeoffs. Software Engineering Institute, Carnegie Mellon University

Maryam, H. (2008). The Evaluation of Reliability Based on the Software Architecture in Neural Networks.

Muhammad Ali, B. (2006). The Value of Architecturally Significant Information Extracted from Patterns for Architecture Evaluation: A Controlled Experiment.

Muhammad Ali, B. and G. Ian (2004). Comparison of Scenario-Based Software Architecture Evaluation Methods. Proceedings of the 11th Asia-Pacific Software Engineering Conference, IEEE Computer Society.

Muhammad Ali, B. (2009). Software Architecture Review: The State of Practice. G. Ian. **42:** 26-32.

Nicholas, M. (2004). A Survey of Software Architecture Viewpoint Models. RMIT University, Melbourne, Australia.

Organisation for Economic Co-operation and Development (OECD). Retrieved January 11, 2010, from http://www.oecd.org

Papazoglou, M. and W.-J. van den Heuvel (2007). "Service oriented architectures: approaches, technologies and research issues." The VLDB Journal **16**(3): 389-415.

Perolof, B., Nico, L., Jan, B., and Hans van, V. (2004). "Architecture-level modifiability analysis (ALMA)." Elsevier Science Inc. **69**: 129-147.

Philippe, K. (1995). The 4+1 View Model of Architecture. **12:** 42-50.

Philippe, K. (2001). Describing Software Architecture with UML.

Philippe, K. (2009). Software Architecture Knowledge Representation [PowerPoint slides]. Retrieved from Software Architecture Representation website: pkruchten.wordpress.com/talks/

Razavi, A. (2010). Software Architecture [PowerPoint slides]. Retrieved from University of Waterloo website: http://courses.softlab.ntua.gr/softeng/Tutorials /SoftArch.pdf

Rowell, B. (2009). Academic Program Assessment. Retrieved April 29, 2010, from oeas.ucf.edu/doc/acad_assess_handbook.pdf

Sadoski, D. (1997). Three Tier Software Architecture. SEI Software Technology Review.

Savola, R. (2006). A Requirement Centric Framework for Information Security

Selim, B. R., Pet-Armacost, J., Albert, A. and Krist, P. S. (2008). Program Assessment Handbook: Guidelines for Planning and Implementing Quality Enhancing Efforts of Program and Student Learning Outcomes, University of Central Florida.

Short, P. M. (2008). Conducting Academic Audit Tennessee Board of Regents: A Case Example [PowerPoint slides]. Retrieved from Tennessee Board of Regents website: http://www.acenet.edu/Content/NavigationMenu/WhatsHot/Short_Paula_07Presentation.ppt

Software Engineering Institute. Retrieved January 11, 2010, from http://www.sei.cmu.edu

The Official Website of Qualifications Agency (MQA). Retrieved January 11, 2010, from http://www.mqa.gov.my

University Malaysia Perlis (UniMAP) Official Website. Retrieved January 11, 2010, from http://www.unimap.edu.my

Wikipedia, The Free Excyclopedia. Retrieved January 11, 2010, from http://en.wikipedia.org

Zhu, L., A. Aurum, et al. (2005). "Tradeoff and Sensitivity Analysis in Software Architecture Evaluation Using Analytic Hierarchy Process." Software Quality Journal **13**(4): 357-375.

Zuhairi, A. and Suparman, A. (2002). Managing quality and accessibility in open and distance learning. In Dikshit, H. P., Garg, S., Panda, S. and Vijayshri (Eds.), Access and equity: challenges in open and distance learning: 252-263.